Toward a Trace-Based PROMETHEE II Method to answer " What can teachers do? " in Online Distance Learning Applications

Hoang Nam Ho, Mourad Rabah, Samuel Nowakowski, Pascal Estraillier

Toward a Trace-Based PROMETHEE II Method to answer "What can teachers do?" in Online Distance Learning Applications

Hoang Nam Ho[1], Mourad Rabah[1], Samuel Nowakowski[2], Pascal Estraillier[1]

[1] L3i Laboratory – University of La Rochelle – La Rochelle – France
[2] University of Lorraine – LORIA, UMR 7503 – Nancy – France

{hoang_nam.ho, mourad.rabah, pascal.estraillier}@univ-lr.fr, samuel.nowakowski@loria.fr

Abstract. In an Online Distance Learning (ODL) application, the learning scenario is a *situation* chaining from an initial system's state to one of the final states where course's criteria are met. A *situation* is a contextualized sequence of interactions. Teachers are successively involved in different *situations*: at the end of one *situation*, they choose the next one. An Intelligent Tutoring System will be used to support teachers in choosing an appropriate situation. In this paper, we show how to apply a decision method (PROMETHEE II) combining with the system traces analysis to automatically recommend to teachers one situation in order to execute during an e-Learning course. Experimental results of our ODL environment are presented to illustrate our method.

Keywords: *Intelligent Tutoring System, PROMETHEE II, traces, situation.*

1 Introduction

Computer science research has been shown to be effective in e-Learning context to increase the students' performance and their motivations. With advancement of artificial intelligence and in cognitive research on human learning, the next generation of computer-based learning moved to develop an Intelligent Tutoring System (ITS). These ITSs are called cognitive tutors that must be able to achieve three main tasks: improve the student's knowledge level, decide what to do next/ adapt instruction accordingly and provide feedback [1]. Given the promising ITS's performance, we interest in analysing how to integrate ITS into e-learning platform to help teacher and also the students during a course. We have developed a case study from our Online Distance Learning (ODL) environment POLARIS [2]. POLARIS is an online blended learning adaptive and interactive service platform. It simulates a virtual classroom with teacher and students' roles for online courses. In this context, we propose to handle this type of application with a hypothesis for adapting its execution according to user behaviours and the current context. We rely on the notion of *situation* [2] to structure the learning scenario. A *learning situation* is a component where the teacher and students interact using local resources associated within a common context to achieve one or more identified course's criteria. The learning scenario in POLARIS is contextualized by the following set of situations: *Presentation*, *Individual Work*, *Group Work*, *Discussion*, *Go to the Board* and *Test*. Thus, teachers execute and participate in successive situations until they reach all of the defined criteria. For each course, the teacher also wants to reach two criteria that are: *Time* (the course must finish on time) and *Comprehension* (the students' comprehension). Normally, after finishing one situation, teachers have to select another one among a set of available situations to carry on the learning course continuation. However, the fact is that teachers are not able to choose precisely what to do in the next execution. Our goal is to integrate an ITS as a teacher assistant to suggest an optimized situation to execute in increasing some of given criteria. In other words, we try to answer the question "What can teachers do?" and find the suggested hint being suitable for students and teacher in order to reach two defined criteria. This question will rely on the choice that will be based on the heuristic multi-criteria decision. Among the many works on multi-criteria decision, we are particularly interested in the PROMETHEE II method [3] that we improve by incorporating a system for analysing the e-learning traces [4] generated during the course session. This is the main method, named Trace-Based PROMETHEE II, which will be integrated into the ITS of POLARIS.

2 Principle of Trace-Based PROMETHEE II

We present the Trace-Based System (TBS) that aims to manage system execution traces. We start by introducing TBS related concepts used in this paper [4]. A *trace* is a sequence of data generated by any action regarding an event occurring during system's execution. Each trace can be associated to a model, called *trace model* that formally represents the corresponding traces. A *modelled-trace* (called *m-trace*) is a trace associated with its model. A system, which manages m-traces, is called *m-Trace-Based System* (m-TBS). The collection phase is devoted to traces collecting. All the m-traces in this phase are called primary traces, defined by $PrT = (S, E, \Omega, C, F)$ where $S = \{si\}$ is a set of executed situations; $E \subseteq S \times S$ is a set of the situations transitions; $\Omega \subseteq V \times S$ is the set of the states of the overall system V and the associated executed situation; C is the set of criteria defined by (*criterion_name, criterion_value*) and F is the teacher's feedback about the obtained criteria when he finished a course with $F = true$ (all of criteria are reached), otherwise $F = false$. The transformation phase selects and transforms the primary traces into new format according to our context. We need a trace that contains a good strategy that means we select all of the primary traces in which the value of F is true. From each obtained primary trace, we have several transformed traces, denoted by $TrT = \Omega$. The analysis phase aims to analyse all of the transformed traces. We then present this phase as the main contribution of Trace-Based PROMETHEE II method.

The teacher has two criteria to reach during his learning session that are: *Time* and *Comprehension*. During the course in POLARIS, we have the state vector V that contains 9 properties such as: *remainT* (the remained time of the course), *presentationT* (the executed time of the situation Presentation), *indiworkT* (the executed time of the situation Individual Work), *groupworkT* (the executed time of the situation Group Work), *discussT* (the executed time of the situation Discussion), *goboardT* (the executed time of the situation Go to the Board), *testT* (the executed time of the situation Test), *questionNum* (the number of students' questions), *questionStu* (the number of students having questions). Our method will analyse after finishing one situation the state vector *V* to choose the most suitable situation among *6* defined situations.

Each criterion is evaluated by one or several properties in the state vector. We decompose the vector *V* into two sub-vectors in terms of two criteria: $v_{time} = (remainT, presentationT, indiworkT, groupworkT, discussT, goboardT, testT)$ and $v_{comprehension} = (questionNum, questionStu)$. We consider the transformed traces base *TrT* in which each transformed trace has the following format: *(V, executed_situation)*. We decompose also the base *TrT* into two sub-bases in terms of two criteria (TrT_{time} and $TrT_{comprehension}$). For each sub-base (containing q records), we extract in *V* all of the properties that contribute to evaluate the correspondent criterion. For each criterion, we compute the Euclidian distance between the state sub-vector and each transformed trace. The value of $dis^{q}_{time}(situation_i)$ represents the distance of the situation *i* between v_{time} and the q^{th} transformed trace in TrT_{time}. We get for each criterion a matrix of size $q \times 1$ containing all the distances between the current state sub-vector and q records in each transformed traces sub-base. The result is the set of two matrixes that correspond to two criteria. From the two matrices above, we choose the *k* smallest distances to compute the evaluation of each situation for each criterion, denoted $E_h(situation_i)$. We choose *k* smallest distances because it represents *k* similar states in the past and for each state, teachers have chosen different situation to execute. Among *k* choices, we compute the probability for each choice in the past. Based on these probabilities, we will obtain the evaluation value of each situation by:

$$E_h(situation_i) = \frac{e_h(situation_i)}{\sum_i e_h(situation_i)} \ with \ e_h(situation_i) = \sum_{j=1}^{k} dis_h^j(situation_i)\,, k \leq q$$

Once these calculations done for each criterion, the PROMETHEE II process continues to build a priority list of situations. The first situation of this list will be the chosen one and suggest that the teacher continue the course. The result of this method can answer our main question of the article "What can teachers do?"

3 Results and Discussion

Our experiments focus on two questions. Does Trace-Based PROMETHEE II suggest a situation to execute? Is this choice satisfied the teacher? We now illustrate the test to answer these questions to confirm the performance of our proposed method in POLARIS platform. We have obtained a base of 1024 transformed traces. From this base, we decompose into 2 sub-bases corresponding to 2 criteria (*Time* and *Comprehension*). We have tested our method in four cases corresponding to four traces bases as described in Fig. 1. We have four traces bases with different number of traces such as: a base containing 50 traces, 100 traces, 500 traces and 1024 traces.

In this test, we use the number of teachers' feedback to evaluate the effectiveness for each traces base. We have totally observed 76 decision-makings using our method. We notice that the more the number of traces we have, the more the number of positive feedbacks we get. We can conclude that the combination between traces and decision method establish a new ITS. This one not only suggests that teacher do but also increases the teacher's satisfaction.

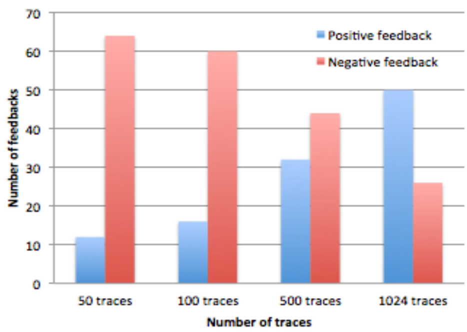

Fig. 1. Impact of number of traces on the result

Table 1. Result of feedbacks between novice and experimented teachers

	Novice teachers		Experienced teachers	
	Course with ITS	Course (not ITS)	Course with ITS	Course (not ITS)
+ feedback	48	11	21	40
- feedback	2	39	29	10

Our method has not always obtained the high positive feedbacks from different teachers as the Table 1. We realize that the ITS integration in a course demonstrates its performance in comparison with a course without ITS for the novice teacher. We observe that the number of positive feedbacks for novice teachers is better with ITS integration than without ITS. In contrast, the positive feedback in course without ITS is higher than that is on the course with ITS for the experienced teachers. The reason is that experienced teachers are better trained in the course. Another limitation is that our method supposes the presence of a configured and up-to-date Trace-Based System. We must have enough data to compute the priority situations list to recommend to teachers. Our future work focuses on integrating the recommendation in our current ITS to improve the inconvenient about the effectiveness of experienced teachers.

References

1. Chakraborty, S., Roy, D., Basu, A.: Development of Knowledge Based Intelligent Tutoring System. Adv. Knowl. Based Syst. …. 1, 74–100 (2010).
2. Trillaud, F., Pham, P.T., Rabah, M., Estraillier, P., Malki, J.: Situation-Based Scenarios for E-learning. IADIS e-learning 2012. pp. 121–128. , Lisbon, Portugal (2012).
3. Taillandier, P., Stinckwich, S.: Using the PROMETHEE multi-criteria decision making method to define new exploration strategies for rescue robots. IEEE International Workshop on Safety, Security, and Rescue Robotics. pp. 193–202 (2011).
4. Cordier, A., Lefevre, M., Champin, P.-A., Georgeon, O.L., Mille, A.: Trace-Based Reasoning - Modeling Interaction Traces for Reasoning on Experiences. FLAIRS Conference. pp. 363–368. AAAI Press, Florida, USA (2013).